

基于软件定义的可重构卷积神经网络架构设计

李沛杰¹, 张丽¹, 夏云飞², 许立明²

(1. 信息工程大学, 河南 郑州 450001;

2. 天津市滨海新区信息技术创新中心, 天津 300457)

摘要: 为满足卷积神经网络业务处理的灵活性和高性能需求, 提出一种基于软件定义的可重构卷积神经网络架构。该架构采用归一化处理流程实现卷积层网络的动态重构与运算模式的加速。采用 AHB 和 AXI 的双总线架构, 实现卷积神经网络的流水计算。通过软件定义在 FPGA 上实现了不同网络结构下的数据集实时处理。实验结果表明, 所设计的 FPGA 电路能够实现两种网络模型的软件定义, 网络模型与输入数据集相同的条件下, 该架构的运算处理能力为 CPU 的 10 倍, 运算能耗比为 GPU 的 2 倍。

关键词: 卷积神经网络; 软件定义; 动态可重构; FPGA; 流水计算; SoC

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2021043

Architecture design of re-configurable convolutional neural network on software definition

LI Peijie¹, ZHANG Li¹, XIA Yunfei², XU Liming²

1. Information Engineering University, Zhengzhou 450001, China

2. Information Technology Innovation Center of Tianjin Binhai New Area, Tianjin 300457, China

Abstract: In order to meet the flexibility and efficiency requirement in convolutional neural network (CNN), an architecture of re-configurable CNN based on software definition was proposed. In the architecture, the process of CNN could be normalized and the operation mode could be accelerated. The calculation pipeline was implemented by using dual bus architecture based on AHB and AXI protocols. By software definition, the proposed architecture, which could realize the real-time processing of data among different CNN structure, was implemented on FPGA. The result shows that at least 2 CNN models can be software defined on the FPGA circuit. The output measures an operation processing capacity of 10 times that of CPU, and an operation energy consumption ratio of 2 times that of GPU.

Keywords: convolutional neural network, software definition, dynamic reconfiguration, FPGA, pipeline calculation, SoC

收稿日期: 2020-09-22; 修回日期: 2020-12-14

通信作者: 李沛杰, lpj@ndsc.com.cn

基金项目: 国家科技重大专项 (2016ZX01012101)

Foundation Item: The National Science and Technology Major Project (2016ZX01012001)

论文引用格式: 李沛杰, 张丽, 夏云飞, 等. 基于软件定义的可重构卷积神经网络架构设计[J]. 网络与信息安全学报, 2021, 7(3): 29-36.

LI P J, ZHANG L, XIA Y F, et al. Architecture design of re-configurable convolutional neural network on software definition[J]. Chinese Journal of Network and Information Security, 2021, 7(3): 29-36

1 引言

随着计算机视觉技术的发展,神经网络凭借深度学习技术极大地提高了算法性能。其中卷积神经网络(CNN, convolutional neural network)作为该项技术的基础,其模型得到了广泛关注,并越来越多地应用于图像识别、视觉拟合、视频搜索等领域^[1]。

CNN 通常由卷积层、全连接层、池化层以及激活层组成^[2-4]。当前 CNN 模型对卷积层与子抽样层互联/迭代层数、卷积核运算尺寸、激活层函数处理方式等处理环节进行不同的优化^[5]。AlexNet 模型以 LeNet-5 模型为蓝本,通过增加卷积层进一步提高模型图像识别率^[5]; VGG 模型通过使用较小卷积核模拟更大尺寸卷积核的方式,在保证层级结构统一且简单的基础上,进一步增加了模型运算深度,提高识别精度^[6];而 GoogleLetNet 模型则通过设计不同尺寸并行卷积模块,并配合传统的池化模块与卷积模块的拼接,在降低运算量与参数量的基础上,仍然保持图像的高识别率^[7-8]。由于 CNN 算法海量数据搬移以及密集迭代运算的特征,通用 CPU 或 GPU 难以实现高效的处理能力, CNN 算法运算数据流的流水线处理特征使 AISC 或 FPGA 等硬件电路能够实现更好的算法加速效能^[9]。专用 ASIC 电路的刚性结构使其本身缺乏算法配置能力,难以应对 CNN 算法快速迭代的需求^[10]。通用的 FPGA 电路实现的 CNN 电路,虽然能够解决算法的迭代更新^[11-13],文献[14]更是基于 FPGA 提出 CNN 优化方案,提升 FPGA 本身在带宽和功耗上的利用率,但 FPGA 的可编程特性属于静态可重构,电路重构时间长,依然难以满足 CNN 算法实时衍进更新的需求^[9]。

本文通过研究卷积神经网络的数据结构特点,采用软件定义互连技术将不同算法中的原子算粒软件定义,构建归一化的算粒集,通过软件定义算粒与数据交换构建算粒互连网络,设计了一种基于软件定义的卷积神经网络可重构电路,同时利用多通道 DMA 高效实现不同搬移规律的二维数组展开。在保持高性能运算处理能力的前提下,基于软件定义实现针对不同卷积神经网络

模型的快速重构,满足卷积神经网络灵活性和高效能的需求。

2 软件定义可重构 CNN 分析

以图像处理应用为例, CNN 算法将图像作为输入,交替迭代进行卷积层特征提取与池化层子抽样得出特征图,最终经全连接层运算得出结果。

卷积层主要完成特征提取运算,处理原理类似数字滤波器,以大小为 $M \times N$ 的卷积核为计算核心,对目标图像进行滑窗点乘遍历,算粒抽象如式(1)所示。

$$X \otimes Y = \sum_{i=1}^M \sum_{j=1}^N X(i, j) \cdot Y(i, j) \quad (1)$$

池化层负责数据抽样处理,实现对特征的深度抽象,减少特征图数据维度。池化处理主要分为均值池化与最大值池化,即对特征图中选定的均分区域 ($N \times N$) 进行均值计算或最大值计算,其处理算粒如式(2)所示。

最大值池化

$$A^{(l)}(i, j) = \max [I_{K \times K}^{(l)}(i, j)] \quad (2)$$

均值池化

$$x_j^l(i, j) = f \left(\frac{1}{n} \sum_{i \in M_j} x_i^{l-1} + b_j^l \right) \quad (3)$$

激活层完成数据的输出控制,即阈值判决选通操作。在 CNN 中常见的激活函数包括纠正线性函数 ReLU、双曲正切函数 tanh、S 形 sigmoid 函数以及符号函数等^[15],运算过程为在激活函数上对将激活前的输出值进行乘叠加后输出,常见的激活函数公式如表 1 所示。

表 1 常见的激活函数公式
Table 1 The typical formulas of activation functions

函数名称	激活函数公式
纠正线性函数 ReLU	$\text{ReLU}(x) = \max(0, x)$
S 形 sigmoid 函数	$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$
双曲正切函数 tanh	$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
符号函数	$\text{sign}(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases}$

全连接层处于 CNN 的最后处理部分，通过将特征图逐行读入并与权重参数矩阵相乘后，完成图像的最终分类处理，其处理算粒如式(4)所示。

$$A^{(l)}(j) = B^{(l)}(j) + \sum_{i=1}^{N_m^{(l)}} B^{(l)}(i) \cdot W(i, j) \quad (4)$$

尽管不同的 CNN 算法处理流程与算粒组合各不相同，但 CNN 算法的流程相对固定，如图 1 所示的 CNN 处理流程中， $K \times K$ 大小的卷积核构成 CNN 运算核心，以不同步长 S 与 Y 类型的激活函数配合，实现卷积层处理；再配合 Q 个下采样数 W 的池化层，堆叠出 J 深度的连续卷积处理；最终与 K 个全连接层互联，完成整个 CNN 流程处理。在各层算粒处理上，算粒具有固定的运算格式，可通过算粒分解实现归一。

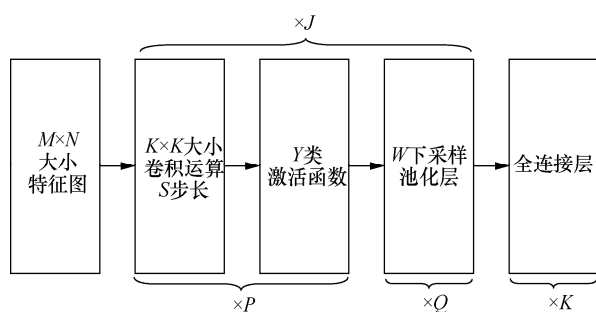


图 1 卷积神经网络处理流程
Figure 1 The process of CNN

本文主要通过抽象 CNN 处理流程中各个算粒的实现方式，通过参数化的处理进行抽象与归一，利用算法流水线处理的特性，将各层算粒归一化为相同的可重构算粒，并将算粒接口进行软件定义化设计，构建软件定义算粒集，同时基于 AHB 和 AXI 双总线构建的互连网络，通过基于路由寻址的数据交换网络实现各个算粒之间数据的传输与互联。

3 软件定义可重构 CNN 电路设计

3.1 总体架构设计

根据上文对卷积设计网络模型的分析，CNN 涉及的运算种类相对固定，访存地址规律固定，数据结构规整，整体运算与数据可重用率高。本文主要基于卷积核尺寸、卷积滑窗步长、卷积迭代层数、池化下采样抽取尺寸以及激活函数类型等功能特性，进行基于软件定义的通用性设计，

以增加部分迭代流程为代价，归一出一种适应多种 CNN 的电路结构，其总体结构如图 2 所示。

其中，软件定义可重构 CNN 整体架构分为两部分：基于 AHB 和 AXI 的双总线架构和多 DMA 引擎的算粒互连网络；基于数据流交换网络和软件定义算粒集合的数据交换网络。整体 CNN 处理流程可通过处理器的软件配置实现算粒互连网络拓扑和数据交换网络拓扑的软件定义动态可重构，满足不同 CNN 算法的灵活配置。在软件定义化的算粒互连网络中，通过高性能 DMA 与通用 DMA，将数据缓存中不同尺寸的二维特征图与卷积核参数转换为一维数据流，并分别通过软件定义数据交换网络与配置总线进入不同的软件定义算粒模块，完成卷积运算；软件定义化的数据交换网络采用基于路由的全交换结构，实现任意软件定义算粒以及高速 DMA 之间的数据流互联。基于该架构，可以针对不同的 CNN 模型的处理流程，软件定义数据流路由，以及配置各软件定义算粒工作模式，处理器启动各 DMA 进行数据流搬移，经过数据流迭代搬移，实现深度卷积层处理。

3.2 软件定义算粒设计

针对卷积神经网络而言，其主要运算类型相对固化，主要的计算算粒总结如下。

- (1) 乘法：两个一维向量数据流相乘运算，用于卷积运算中乘法以及因子缩放。
- (2) 加减法/累加：实现卷积运算中累加运算，也可以实现池化运算中偏置常量加法。
- (3) 比较：一维向量数据比较结果输出或计算置位非等价操作，可以实现激活函数或池化采样等处理。
- (4) 最值统计：求取一维向量中最大/小值，可实现 ReLU 激活函数操作。
- (5) 移位：通过对数据的左移/右移，实现输入数据的 2 的幂次方运算，实现数据缩放，节省乘法资源。
- (6) 向量复制：将输入的一维向量复制多出一路，实现向量平方运算，用于可能的超越运算泰勒级数展开处理。
- (7) 序列生成：针对输入的常数，展开为任意长度的一维数据流，实现矩阵常数乘法。
- (8) 数据直通：将输入的数据不做任何处理，

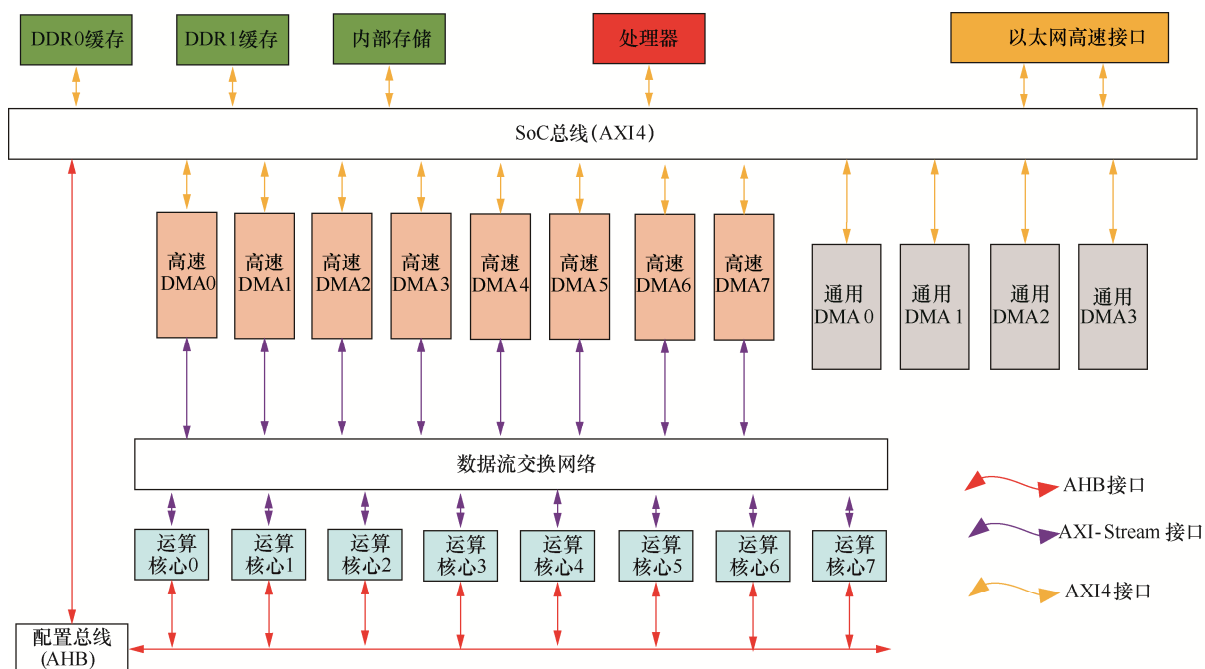


图 2 软件定义可重构 CNN 电路架构
Figure 2 The architecture diagram of re-configurable CNN on software definition

直接透传到下一级运算核心。

计算算粒的软件定义互连接口按照 AHB 和 AXI 的标准协议接口进行归一化, 运算核心整体数据流采用基于握手的流水线处理, 所有算粒的输入输出均与数据网络互联, 配合数据交换网络完成核心数据流交换。针对以上所有处理进行分类, 将移位、向量复制、序列生成等处理统一放入运算核心数据入口处作为数据预处理模块; 将乘法与加减法/累加等操作作为运算处理模块归为一类; 将最值统计与比较操作作为后处理子模块实现, 整体模块框图如图 3 所示。

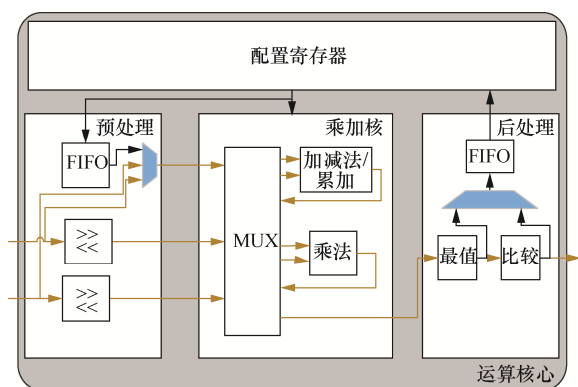


图 3 软件定义算粒框图
Figure 3 The diagram of software defined cell

运行算粒模块主要划分为 4 个子功能模块: 预处理模块、乘加核模块、后处理模块以及配置寄存器模块, 各模块主要功能如下。

1) 预处理模块: 负责移位、复制以及序列生成等功能, 子模块为 2 输入 3 输出设计, 新增数据路径用于传输复制或常数序列。内部设计下行 FIFO, 通用 DMA 可以通过 AHB 配置总线向 FIFO 中自动写入数据, 以实现将 FIFO 参数进行常数展开。移位操作可以通过配置为 0 移位实现数据透传。

2) 乘加核模块: 内部设计全互联数据 MUX 模块, 可以通过配置实现数据流通过 2 个运算、1 个运算或直接透传, 并且可以决定进入乘法与加法运算核的先后顺序, 以实现不同算法公式; 同时乘法与加法模块实现采用单精度浮点运算。

3) 后处理模块: 主要完成对任意尺寸的一维输出数据的最值提取, 以及比较后降采样操作, 并且设计上行 FIFO, 通用 DMA 可通过配置总线读取 FIFO 中最值或比较值结果, 上送 SoC 总线后缓存至内部存储中。

4) 配置寄存器模块: 实现对运算核心的工作模式, 实现对一维数据尺寸、移位位数、加减法/

累加、最大/小值、数据复制、比较阈值等模式的切换, 实现运算核心的重配置。

整个运算核心采用 2 数据流输入 1 数据输出接口结构, 所有输入输出接口均与数据交换网络相连, 数据来源为高速 DMA 从 DDR 中读出的数据缓存。数据流输出接口附带目前模块路由信息, 用于数据交换网络的路由寻址。同时, 内部设计与配置寄存器相连的上行与下行 FIFO, FIFO 与通用 DMA 之间设计握手连接信号, 通过通用 DMA 对 FIFO 状态的动态感知调节数据流量, 从而实现常数数据的批量一维展开和各种卷积神经网络的运算处理。

3.3 软件定义 DMA

本文中的高速 DMA 与通用 DMA 是数据搬运核心, 完成数据从 SoC 总线上存储设备与运算核心之间的数据搬运。软件定义可重构 CNN 架构中内置一个处理器, 用来管理两种 DMA 的指令队列, 用于 DMA 内数据通道的配置, DMA 功能总结如下。

- 1) 所有 DMA 均可以通过 SoC 总线访问总线上所有存储。
- 2) 所有 DMA 内置指令队列, 用来定义源地址、目的地址、数据长度、burst 个数、地址变化方式等描述符任务。
- 3) 所有通用 DMA 支持 4 个逻辑通道同时数据搬运。
- 4) 高速 DMA 支持固定地址、顺序地址、跳跃地址、跳跃片段地址等地址跳跃形式。

高速 DMA 内部针对读写各有一个数据通道, 实现复杂的地址循环, 将 SoC 总线读取的数据直接转化为数据流送入数据交换网络中, 完成特征图数据与卷积核参数的批量读取; 而通用 DMA 中采用多逻辑通道时分复用的形式, 完成低速参数的自动搬运。工作过程中两种 DMA 同时被处理器启动, 高速 DMA 将一维化特征图数据经过总线与数据交换网输出到运算核心, 与通用 DMA 输出的参数或其他一维化特征图进行运算, 实现二维数据的流水式运算。

3.4 数据交换网络

数据交换网络主要完成软件定义算粒、软件定义 DMA 等运算处理模块的交换互联, 实现点

对点之间数据流的交互, 软件定义互连接口采用 AXI4-Stream 协议, 可通过整个网络接口之间的握手机制形成流控。AXI4-Stream 协议接口中分为 master 接口与 slave 接口, master 接口为输出接口, slave 接口为输入接口, 两种接口互为对称衔接。数据交换网络对所有 master 接口进行编址, 从 slave 接口输入对应的路由地址, 即可实现网络内对应 slave 接口到 master 接口之间的路由。

数据交换网络根据不同算法流程规划, 运算核心个数设置为 8 个, 同时每个运算核心都是 2 输入 1 输出的端口数量, 合计 master 接口数量 16 个, slave 接口数量 8 个, 计算 8 个高速 DMA 的 AXI4-Stream 接口中, master 与 slave 各 8 个接口。整个交换网络的需要 24 个 master 接口, 以及 16 个 slave 接口。由于采用的总线 IP 最大支持 16×16 规模交换网络, 所以采用两级拓扑实现 16×24 数量接口的数据网络。如图 4 所示, 每个 master 接口均被编址。

4 FPGA 实现与结果分析

4.1 FPGA 实现

本文进行硬件设计的开发平台为 Xilinx 公司的 ZC706 开发板, 其 FPGA 芯片包含 PS 和 PL 两种硬件资源, 利用其 PS 内的 ARM 双核处理器、AXI4 总线以及 SDMA 等 IP 分别实现本文中处理器、SoC 总线以及通用 DMA 功能; 在 PL 部分实现运算核心、数据交换网络与高性能 DMA。测试集为 MNIST 集合中 200 幅 256×256 大小的图片, 每个 Pixel 的取值是 $0 \sim 255$ 的整数, 在进行实现前将所有像素转换为单精度浮点数后进行实验测试。测试算法包括 AlexNet 与 VGG-16 两种不同算法, 分别在本文电路架构上实现运算, 使用相同算法与输入数据集分别在通用处理器 Intel CORE i5-7200U 与 NVIDIA GTX1050 GPU 上进行处理时间的性能对比, 并且 CPU 与 GPU 采用与本设计电路相同的数据精度, 均为单精度浮点数据。

4.2 实验结果

本文设计中的 FPGA 芯片内部处理主频为 200 MHz, 在 FPGA 中实现的卷积神经网络电路

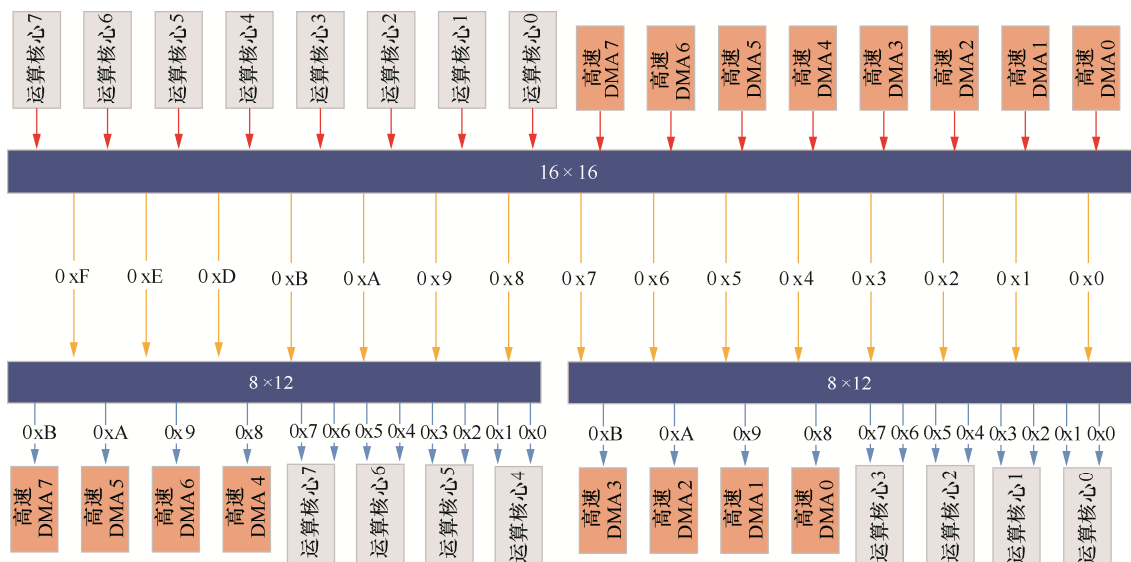


图 4 数据交换网络拓扑

Figure 4 The topology of data switching network

可以通过 ZYNQ 中 ARM 处理器的配置, 完成电路的动态重构。所以在切换不同网络模型时, FPGA 无须改变电路结构, 可以采用与 CPU 或 GPU 相同的方式直接切换不同工作软件。表 2 为整个 FPGA 芯片资源使用情况。

表 2 FPGA 芯片资源使用情况
Table 2 The usage of FPGA resource

资源类型	资源数量	占用比例
LUT	188 831	85.17%
Flip-Flops	258 938	59.22%
DSP48E	804	89.24%
Block RAM	677	62.03%

表 3 为 3 种运算平台使用 AlexNet 网络模型的对比结果, 针对 200 幅 256×256 大小的单精度浮点图片进行运算。经过对比可以看出, 本文所设计的软件定义可重构 CNN 处理电路是通用 CPU 处理速度的近 10 倍, 但相比 GPU 并无性能优势, 其主要原因是 GPU 的运算速度相比 FPGA

更快, 考虑到频率差异, 本文设计在性能功耗比上占据一定优势, GPU 能耗比约 $200 \div 1.722 \div 300 \approx 0.387 1$ (FPS/W), 本文所设计的软件定义可重构 CNN 电路性能功耗比为 $200 \div 31.84 \div 7.8 \approx 0.8053$ (FPS/W), 约为 GPU 的 2.08 倍。

切换 3 个平台网络模型处理软件, 运行 VGG-16 网络模型算法, 并对 100 幅 256×256 大小的单精度浮点图片进行运算, 3 个平台处理算法速度对比如表 4 所示。3 个平台处理效率比率结果和变化趋势与 AlexNet 网络模型算法的处理相同, 并且本文设计 FPGA 平台实现了软件配置切换实现不同卷积神经网络算法。

由 VGG-16 网络模型算法和 AlexNet 模型算法运行情况的性能分析结果可知, 本文所设计的基于软件定义的可重构卷积神经网络架构相比通用 CPU 与 GPU 有可观的性能功耗比优势, 其主要原因是采用了基于数据流的运算网络结构, 在每次的运算过程中数据流进行逐级流水运算, 每

表 3 AlexNet 模型算法效率对比
Table 3 The efficiency comparison of AlexNet algorithm among different platform

平台类型	图片数量/幅	处理时间/s	功耗/W	性能功耗比率/FPS·W ⁻¹
Intel CORE i5-7200U	200	310.07	25	0.024 8
NVIDIA GTX1050 GPU	200	1.722	300	0.387 1
本文设计软件定义 CNN	200	31.84	7.8	0.805 3

表4 VGG-16 模型算法效率对比
Table 4 The efficiency comparison of VGG-16 algorithm among different platforms

平台类型	图片数量/幅	处理时间/s	功耗/W	性能功耗比率 (FPS/W)
Intel CORE i5-7200U	100	252.55	25	0.015 8
NVIDIA GTX1050 GPU	100	1.412	300	0.236 1
本设计软件定义 CNN	100	26.10	7.8	0.491 2

次运算时间等于数据访问存储时间加流水运算时延,并没有传统运算器的额外运算开销,所以可以实现高速的流水并行运算,整个系统针对固定的卷积算法模型具有固定的时延,不会因业务流的大小产生偏差,整体性能取决于系统工作频率,可通过硬件电路的升级和工作频率的提升,进一步提升处理性能。此外,软件定义可重构的 CNN 能够实现基于软件配置的动态可重构,实现多种算法模式的动态切换,及基于软件定义算粒的算法迭代与更新,具备较高的算法泛用性。

5 结束语

本文基于软件定义技术,通过构建归一化的软件定义计算算粒和数据交换网络模型,提出了一种 CNN 算法泛化的可重构的卷积神经网络架构,通过可编程 DMA 配合可重构软件定义算粒核心实现动态可重构多种 CNN 算法的目的。通过 FPGA 实现了该架构下的两种典型算法模型,其能够满足卷积神经网络算法快速迭代和实时处理的需求。后续会进一步整合并添加其他新型 CNN 算法模型运算核心、提高本电路的应用普适性,同时继续优化架构,在保持架构灵活配置的基础上在 ASIC 专用电路上进行实现,设计卷积神经网络领域专用的高效能计算芯片。

参考文献:

- [1] SINGH K, TIWARI S C, GUPTA M. A closed-loop ASIC design approach based on logical effort theory and artificial neural networks[J]. *Integration*, 2019, 69: 10-22.
- [2] LIU Z, DOU Y, JIANG J, et al. An FPGA-based processor for training convolutional neural networks[C]//International Conference on Field Programmable Technology (ICFPT). 2018.
- [3] 方睿, 刘加贺, 薛志辉, 等. 卷积神经网络的 FPGA 并行加速方案设计[J]. *计算机工程与应用*, 2015, 51(8): 32-36.
FANG R, LIU J H, XUE Z H, et al. FPGA-based design for convolutional neural network[J]. *Computer Engineering and Applications*, 2015, 51(8): 32-36.
- [4] 王巍, 周凯利, 王伊昌, 等. 卷积神经网络(CNN)算法的 FPGA 并行结构设计[J]. *微电子学与计算机*, 2019, 36(4): 57-62, 66.
WANG W, ZHOU K L, WANG Y C, et al. FPGA parallel structure design of convolutional neural network(CNN) algorithm[J]. *Microelectronics & Computer*, 2019, 36(4): 57-62.
- [5] 翟社平, 邱程, 杨媛媛, 等. 基于 FPGA 的卷积神经网络加速器设计与实现[J]. *微电子学与计算机*, 2019, 36(8): 83-86.
ZHAI S P, QIU C, YANG Y Y, et al. Design and implementation of convolutional neural network accelerator based on FPGA[J]. *Microelectronics & Computer*, 2019, 36(8): 83-86.
- [6] 窦阳, 卿颢波, 何小海, 等. 基于 FPGA 的 CNN 加速器设计与实现[J]. *信息技术与网络安全*, 2019, 38(11): 96-101.
DOU Y, QING L B, HE X H, et al. Design and implementation of CNN accelerator based on FPGA[J]. *Information Technology and Network Security*, 2019, 38(11): 96-101.
- [7] 周琰, 张铃. 模糊集方法在检索评价系统中的应用[J]. *计算机技术与发展*, 2007, 17(1): 111-113.
ZHOU Y, ZHANG L. Application of fuzzy measure in information retrieval evaluation[J]. *Computer Technology and Development*, 2007, 17(1): 111-113.
- [8] VENUGOPAL S, CASTRO-PAREJA C R, DANDEKAR O S. An FPGA-based 3D image processor with median and convolution filters for real-time applications[C]//Proc of Spie-is&t Electronic Imaging. 2005.
- [9] CHEN J, PRODIC A, ERICKSON R W, et al. Predictive digital current programmed control[J]. *IEEE Transactions on Power Electronics*, 2003, 18(1): 411-419.
- [10] CHELLAPILLA K, PURI S, SIMARD P. High performance convolutional neural networks for document processing[C]//10th International Workshop on Frontiers in Handwriting Recognition. 2006.
- [11] YU Y, WU C, ZHAO T, et al. OPU: an FPGA-based overlay processor for convolutional neural networks[J]. *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, 2019, 28(1): 35-47.
- [12] WU D, PIGOU L, KINDERMANS P J, et al. Deep dynamic neural networks for multimodal gesture segmentation and recognition[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2016, 38(8): 1583-1597.
- [13] 仇越, 马文涛, 柴志雷. 一种基于 FPGA 的卷积神经网络加速器设计与实现[J]. *微电子学与计算机*, 2018, 35(8): 68-72.

QIU Y, MA W T, CHAI Z L. Design and implementation of a convolutional neural network accelerator based on FPGA[J]. Microelectronics & Computer, 2018, 34(8): 68-72.

- [14] 刘勤让, 刘崇阳, 周俊, 等. 基于线性脉动阵列的卷积神经网络计算优化与性能分析[J]. 网络与信息安全学报, 2018, 4(2):16-24.

LIU Q R, LIU C Y, ZHOU J, et al. Based on linear systolic array for convolutional neural network's calculation optimization and performance analysis[J]. Chinese Journal of Network and Information Security, 2018, 4(2): 16-24.

- [15] 梁爽. 可重构神经网络加速器设计关键技术研究[D]. 北京: 清华大学, 2017.

LIANG S. Research on key technologies of reconfigurable neural network accelerator design[D]. Beijing: Tsinghua University, 2017.

[作者简介]



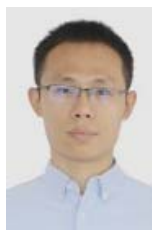
李沛杰(1990-), 男, 山西襄汾人, 信息工程大学助理研究员, 主要研究方向为先进计算与网络安全技术、高速接口技术、软件定义互联技术、现代 SoC 设计技术。



张丽(1982-), 女, 河南辉县人, 信息工程大学副研究员, 主要研究方向为先进计算技术、软件定义互联技术、SoC 芯片设计技术。



夏云飞(1987-), 男, 黑龙江海伦人, 天津市滨海新区信息技术创新中心工程师, 主要研究方向为信号处理算法高性能实现技术、软件定义互联技术、高性能芯片架构设计技术。



许立明(1990-), 男, 山东潍坊人, 天津市滨海新区信息技术创新中心工程师, 主要研究方向为 FPGA 设计实现、芯片设计前端流程以及 SoC 芯片系统时钟架构设计。